

REMARKS/ARGUMENTS

This Amendment and the following remarks are intended to fully respond to the Office Action dated February 27, 2006. In that Office Action, claims 1-13 were examined, and all were rejected. More specifically, claim 7 stands rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which the Applicant regards as the invention; and claims 1-13 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,175,852 to Johnson et al. (hereinafter, "Johnson"), in view of the article "Performance of the IBM General Parallel File System" by Terry Jones, Alice Koniges, and R. Kim Yates (hereinafter, "Jones"). Reconsideration of these rejections, as they might apply to the original and previously amended claims in view of these remarks, is respectfully requested.

In this Response, claims 1, 7, and 11 have been amended to improve their form. No claims have been canceled or added. Therefore, claims 1-13 remain present for examination.

Claim Rejections – 35 U.S.C. § 112

Claim 7 stands rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which Applicant regards as the invention. In particular, the Examiner rejected claim 7 for indefiniteness, stating that the phrase "may be" is deemed broad and indefinite. While Applicant respectfully disagrees that the phrase "may be," as used in the context of the claim language of claim 7, is indefinite, Applicant has amended the claim language of claim 7 to provide further clarification. Applicant has thus amended claim 7 to read, "A method as defined in claim 1 wherein more than one client computer system can lock the resource." By the plain language of claim 7, and when read in light of the specification (see page 7, lines 10-18), a person of ordinary skill in the art would understand that, according to one embodiment of the invention, more than one client computer system can lock a given resource in a distributed environment. Applicant thus respectfully requests that the Examiner reconsider the 35 U.S.C. § 112, second paragraph, rejection of claim 7.

Claim Rejections – 35 U.S.C. § 103

Claims 1-13 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Johnson in view of Jones. Applicant respectfully traverses the § 103(a) rejections because the Examiner has failed to state a prima facie case of obviousness. A prima facie case of obviousness can be established only when all of the following requirements are satisfied: (1) the reference or combination of references must teach or suggest all of the claim limitations; (2) there must be some suggestion or motivation in the references themselves to combine the references; and (3) there must be a reasonable expectation of success. See MPEP §§ 706.02(j) & 2143. Thus, the combination of references cited by the Examiner must teach or suggest every limitation of the claimed invention. CFMT, Inc. v. YieldUp Int'l Corp., 349 F.3d 1333, 1342 (Fed. Cir. 2003); see also MPEP § 2143.03.

The present invention, as defined in the claims, relates generally to distributed computing environments and, particularly, to a method and system for creating and maintaining lock properties for a resource or object in a distributed computing environment. More particularly, the invention relates to locking methods that extend the WebDAV protocol, or the World Wide Web Distributed Authoring and Versioning standard. A distributed environment allows multiple clients to share a computer resource(s). Where a computer resource is shared, an embodiment of the present invention provides for a locking scheme to be used to prevent “lost update” problems from arising where two or more users modify a shared resource at the same time. According to embodiments of the present invention, a determination operation determines whether a requested resource may be accessed. If the resource may be accessed, a lock object is created, in which a lock object comprises information related to whether its associated data object is locked and therefore inaccessible by other client computing systems. Following the creation of the lock, a lock token is returned to the client that requested access to the object, according to an embodiment of the invention. Among other types of information, this lock token provides information to the client relating to the newly created lock object.

While the present invention relates to management of resources in a distributed environment generally and, more particularly, to locking methods that extend the WebDAV protocol, Johnson relates exclusively to management of distributed data processing systems involving the UNIX operating system, or other systems with **similar** characteristics to those of

the UNIX operating system. See col. 1, lines 49-63 (the preferred embodiment of the invention “is implemented in a version of the UNIX operating system; however, the invention could be implemented in other and different operating systems.”); and col. 2, lines 52-55. In particular, Johnson expressly states that such “other systems” must be those with characteristics similar to the UNIX operating system: “The invention to be described hereinafter was implemented in a version of the UNIX operating system but may be used in other operating systems having characteristics similar to the UNIX operating system.” Col. 2, lines 52-55 (emphasis added). In this UNIX-based, or UNIX-type, operating system, the invention provides for locking byte ranges within files so that other processes cannot access those ranges. Col. 20, lines 61-63. The invention provides for two types of locks, namely the exclusive write lock and the shared read lock. Col. 21, lines 1-11. Johnson provides three different UNIX operating system commands related to locking. Col. 21, lines 31-52. Johnson does not relate to the use of lock tokens.

With regard to Jones, while the present invention relates primarily to sharing software-related resources, the Jones article relates exclusively to a hardware-based method of data access that is designed to increase the throughput, and throughput rates, of data by providing for parallel file systems. See Jones, § 2 (“The [General Parallel File System (GPFS)] architecture was designed to achieve high bandwidth for concurrent access to a single file . . . [t]he intended platform for this file system is IBM’s line of massively parallel computers, the RS/6000 SP, and performance is achieved with commodity disk technology. The RS/6000 SP line of machines are general purpose, high[-]end computers which scale to thousands of processors.”). See also Jones at § 2.3 (“[T]he degree of scalability is probably the most unique feature of GPFS. This design permits a file to be striped across a system-administrator-defined number of server nodes. Not only does this provide higher aggregate read and write performance, it also permits larger files and file systems.”). GPFS is implemented as multiple separate software subsystems or services, in which each service may be distributed across multiple nodes. Jones at § 2.1. Among other features, a token manager server is used to synchronize concurrent access to files and ensure consistency among caches. Id.

A. Rejections of Claims 1-12

Johnson and Jones fail to satisfy the first prong of establishing a prima facie case of obviousness because they fail to teach or suggest all of the claim limitations, particularly in light of the new amendments to claims 1 and 11. While the Applicant respectfully disagrees with the Examiner's rejections of claims 1-13 under § 103(a) as being unpatentable over Johnson in view of Jones, the Applicant has amended claims 1 and 11 for purposes of providing clarification of the non-obvious nature of the present invention and to thus further this application to allowance. In light of these amendments, the Applicant respectfully believes that the Examiner's § 103(a) rejections are rendered moot and offers the following discussion for purposes of clarification.

As amended, claim 1 provides for (among other elements) “creating a lock having a predetermined type, wherein the predetermined type provides availability to other client computer systems for predetermined purposes;” (emphasis added), and “returning a lock token upon the creation of the lock to the requesting client computer system.” (Emphasis added.) Similarly, claim 11 provides “if the requesting client computer system does not honor the advisory lock or if the resource is not locked with a conflicting lock, then creating a lock, returning a lock token upon creation of the lock, and performing the access.” (Emphasis added.) As can be seen, each of these elements of independent claims 1 and 11 requires that a lock token be returned to the requesting client upon the creation of a lock object. However, Johnson and Jones fail to teach or suggest the returning of a lock token upon the creation of the lock object. Indeed, the Examiner has expressly stated that “Johnson does not detail the use of tokens.” Examiner's Response to Amendment, Office Action, 2/27/2006, pp. 3-12 (emphasis added).

Further, contrary to the Examiner's arguments, Jones, like Johnson, fails to teach or suggest the return of a lock token to the requesting client upon the creation of a lock object for that client, as required by independent claims 1 and 11. According to an embodiment of the present invention, a lock object is created upon determining that a resource may be accessed. Following the creation of the lock object, a lock token is returned to the client that requested access to the object. Upon returning the lock token to the requesting client, the client may then perform the type of access on the resource that has been requested by the client. Application, p. 20, lines 9-20; see also FIGS. 4 & 5. While the present invention refers to “creating” a lock

object in accordance with an embodiment of the invention, alternative embodiments provide that a lock object is created in parallel with the creation of a data object, or, the lock object may be created and later associated with the data object once the data object is created. Regardless, independent claims 1 and 11 each recite the step of “creating” a lock object (at some point), as well as the return of a lock token to the requesting client.

Jones, on the other hand, does not provide for a lock token to be returned to the requesting client upon creation of the lock object. First, Jones does not disclose “creating” a lock upon request by the client but, instead, refers only to all files as already being considered lock objects (in and of themselves) in the first instance. See, e.g., § 2.1 (“The item being accessed (for example, a file) is termed a lock *object*” (emphasis in original)). Second, Jones does not disclose returning a lock token upon creation of the lock object. Instead, Jones requires that a token manager server hold the “list of nodes that have the [desired] token.” Id. To obtain the desired lock token, “[t]he mmfsd negotiates with the node that holds the token in order to get the requested token. It first contacts the token manager server for a list of nodes that have the token, then it negotiates with the tokens in that list to acquire the token.” Id. Thus, Jones discloses a method and system for maintaining a list of nodes holding certain tokens and the need to “negotiate” to obtain the desired token. Jones in no way provides for the present invention’s return of a lock token to the requesting client upon the creation of the lock object. Instead, Jones relates only to housing a list of nodes corresponding to certain tokens and the need for a requesting client to negotiate in order to obtain one of those tokens. The following example illustrates the fundamental differences between the lock tokens of an embodiment of the present invention and Jones:

Present Invention: Create Lock Object → Return Lock Token → Perform Access

Jones: Lock Object → Obtain List of Nodes Having Token →

Negotiate to Acquire Token → Acquire Token → Perform Access

With respect to the present invention, even where a lock already exists (as with advisory locks in accordance with an embodiment of the invention, for example), see, e.g., FIG. 5, the requested access is either denied altogether or, where the requesting application does not recognize advisory locks, a lock object is created and a lock token is returned to the requesting

application upon creation of the lock object. The present invention thus does not obtain a list of nodes having tokens, nor does the presently recited invention negotiate to acquire a token. Therefore, the present invention's use of lock tokens is conceptually and operationally distinct from that of Jones.

Thus, while Jones uses the terminology of "lock tokens," Jones altogether fails to suggest, much less teach, any process or system wherein a lock token is returned to the client upon the creation of a lock object for the requesting client. Indeed, Jones **teaches away** from returning a lock token to the requesting client upon creation of the lock object by instead teaching a central housing mechanism for nodes and corresponding tokens, as well as the need for a requesting client to contact the central housing server for purposes of negotiating to obtain a desired token. Further, because Jones relates exclusively to a hardware and system-software based method of data access, Jones relates to a distinct field of art from the claimed invention, which relates in embodiments to sharing software-related resources, e.g., application or network software. Accordingly, while Jones may refer to a "lock" and a "token," it inherently must attach a different conceptual meaning to these terms than that attached by the present invention. Any similarities in the selected terminologies of Jones and the claimed invention must therefore be considered in the context of their separate, and distinct, inventions.

For at least these reasons, the Applicant respectfully requests reconsideration of the Examiner's rejection of claims 1 and 11 in view of Johnson and Jones, as these claims are believed to recite the present invention in a manner distinguishable over any combination of the above references. Because neither Jones nor Johnson teaches or suggests the returning of a lock token to the requesting client upon the creation of a lock object, these references fail to satisfy the requirements for establishing a prima facie case of obviousness. Further, the Applicant believes that the current amendments render moot the Examiner's arguments. The Applicant has therefore not addressed the Examiner's arguments that Johnson discloses the remaining elements of claims 1 and 11. However, the Applicant's lack of addressing these arguments should in no way be construed as an indication of acquiescence to, or agreement with, the Examiner's arguments. Because the Applicant believes that claims 1 and 11 are patentable over Johnson in view of Jones,

claims 2-10 and 12 are also believed to be patentable over Johnson in view of Jones as these claims depend directly or indirectly from the allowable base independent claims 1 and 11.

B. Rejection of Claim 13

With respect to claim 13, Johnson and Jones fail to satisfy the first prong of establishing a prima facie case of obviousness because they fail to teach or suggest all of the claim limitations of claim 13. Claim 13 provides, among other elements, that a locked resource comprises, “a lock object, wherein the lock object may comprise one or more of the following properties: nosharewrite, nosharedelete, noshareread, and advisory.”

Because Johnson and Jones relate to entirely distinct fields of art from each other and from the present invention, the Applicant respectfully disagrees that Johnson in view of Jones teaches the present invention’s lock properties of nosharewrite, noshareread, and advisory. See discussion supra. Further, Johnson and Jones are altogether silent as to a lock object having a lock property of “nosharedelete.” Indeed, neither Johnson nor Jones makes any mention of a lock property relating to “nosharedelete,” in which the present invention’s nosharedelete lock property allows other clients to read or write to an otherwise locked application but prevents any deletion operations. In fact, Jones does not provide for any shared locktypes at all. Instead, Jones expressly states: “[I]f two separate nodes write to the same file, and if the writes are overlapping, the overlapped region must be either entirely from node A or entirely from node B.” Id. (emphasis added).

In arguing that Johnson provides for the lock property of nosharedelete, the Examiner points to the language in Johnson that “[w]hen the server deletes a file, it increments the inode generation number.” Col. 14, lines 9-10. However, this language is not related in any way to a lock property of nosharedelete associated with a lock object, as disclosed in accordance with an embodiment of the present invention. When read in the context of Johnson’s entire disclosure, Johnson’s reference to deleting a file and incrementing the inode generation number relates to solving the problem that may arise if a client attempts to perform an operation on a file which has been replaced by another file unbeknownst to the client. Johnson thus has nothing to do with

a lock property associated with a lock object that prevents a client from performing deletions in an accessed resource. Therefore, Johnson in view of Jones in no way suggests, much less teaches, the present invention's lock property of nosharedelete.

Further, the Applicant respectfully disagrees with the Examiner's view that the lock property of nosharedelete is rendered obvious because "[f]ile deletion means are an obvious feature of file systems." Examiner's Response to Amendment, Office Action, 2/27/2006, p. 12. First, the no-deletion lock properties of the present invention's nosharedelete are functionally and conceptually distinct from the mechanism of merely deleting a file. Second, if the Examiner's argument were to hold weight, any invention which involves the mechanism of deleting could be said to be anticipated or rendered obvious by the first piece of prior art which related generally to the deletion of a file, or of any object or entity, in a computing environment. Such clearly cannot be the result intended by the Examiner or by the Patent Office.

Because neither Johnson nor Jones suggests, much less teaches, the "nosharedelete" lock property of claim 13, these references fail to satisfy the first prong of the obviousness test. For at least these reasons, the Applicant respectfully requests reconsideration of the Examiner's rejection of claim 13 in view of Johnson and Jones as this claim is believed to recite the present invention in a manner distinguishable over any combination of the cited references. While in this discussion, the Applicant has focused on the "nosharedelete" property so as to fully respond to the Examiner's arguments, this focus should in no way be construed as indicating that the Applicant believes that the other elements and/or limitations of claim 13 are somehow suggested or taught by Johnson in view of Jones. Rather, because Johnson in view of Jones fails to teach or suggest the nosharedelete property, these references fail to satisfy the first prong of the obviousness test, and arguments related to the other elements and/or limitations are not necessary. Accordingly, reconsideration of the Examiner's rejection of claim 13 is respectfully requested.

Conclusion

This Amendment fully responds to the Office Action mailed on February 27, 2006. It is recognized that the Office Action may contain arguments and rejections that are not directly

addressed by this Amendment due to the fact that they are rendered moot in light of the preceding arguments in favor of patentability. Hence, the failure, if any, of this Amendment to directly address an argument raised by the Examiner should not be interpreted as reflecting the Applicant's belief that such argument, if any, has merit. Furthermore, the claims of the present application may include other elements, not discussed in this Amendment, which are not shown, taught, or otherwise suggested by the art of record. Accordingly, the preceding arguments in favor of patentability are advanced without prejudice to other bases of patentability.

It is believed that no further fees are due with this Amendment. However, the Commissioner is hereby authorized to charge any deficiencies or credit any overpayment with respect to this patent application to deposit account number 13-2725.

In light of the above remarks and amendments, it is believed that the application is now in condition for allowance, and such action is respectfully requested. Should any additional issues need to be resolved, the Examiner is requested to telephone the undersigned to attempt to resolve such issues, if any.

Respectfully submitted,

Dated: 5/30/06



Elizabeth J. Reagan
Elizabeth J. Reagan, Reg. No. 67,528
MERCHANT & GOULD P.C.
P.O. Box 2903
Minneapolis, MN 55402-0903
303.357.1644